Semih Balki-19010

P1)

i)

a = number of men

b = number of women

n = a + b

k = upper bound(n / 2)

<u>Input:</u>

is a 2D matrix of size n * k

Rows from 0 to n - a - 1 represent preference list of men.

Rows from n - a to n - 1 represent preference list of women.

w1	w2	w3	w4	...	...	...	wn/2
w1	w4	w3	wn/2	...	...	...	w20

.

.

.

m4	m3	m1	m8	...	...	...	mn/2

.

.

.

m3	m2	m1	mn/2	...	...	...	m6

mi: represents men where i = 1, 2, …, n/2

wj: represents the women where j = 1, 2, …, n/2

Output: is list of chosen pairs in 2 columns and (n / 2) + 1 rows.

	Women	Men	
w1	m2	w7	m1
w2	m1	w3	m2
w3	m5	w11	m3
.	.	.	.
.	.	.	.
.	.	.	.
wn/2	mn/2	wn/2	mn/2

-From 1st row to n/2 th row at 0th column represents the matches of each women from w1, w2, …, wn/2
-From 1st row to n/2 th row at 1st column represent the matches of each man from m1, m2, …, mn/2

ii)
Suppose there are three women {w1, w2, w3} and three men {m1, m2, m3} whose preferences are shown below, in order from top to bottom

w1	w2	w3
w1	w3	w2
w2	w1	w3
m3	m1	m2
m2	m1	m3
m1	m2	m3

P2)

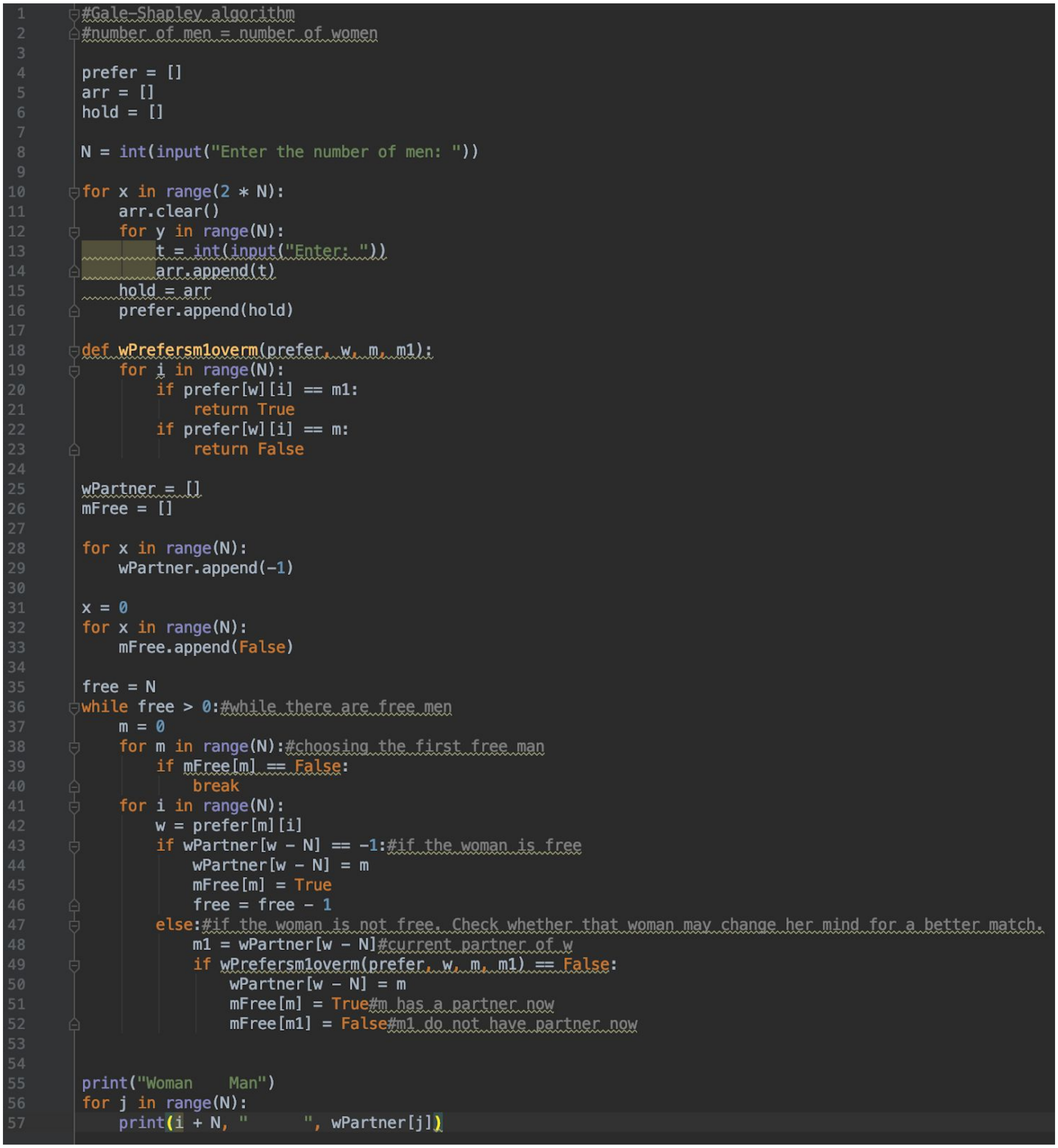
i)

```
free <- n //n: number of free(not married) men
do while free > 0
{
    m <- choosing the first free man
    w <- highest ranked such woman of m to whom he has
        not proposed yet
    if w is free
    {
        (m, w) become engaged //saying ’engaged’, ‘not married’ since
                                //Idea of the woman might change
        Free <- free - 1
    }
    else //check whether that do engagement will remain stable
    {
        if w prefers m to m’
        {
            (m, w) become engaged
            m’ become free
        }
    }
}
```

ii)

O(n^2) since the outer while loop repeats at most n times and (choosing the first free man is at most n times or choosing the highest ranked such woman of m to whom he has not proposed yet is also at most n times) and all the other else is constant so the complexity of the algorithm is O(n^2)

P3)



```
#Gale-Shapley algorithm
#number of men = number of women

prefer = []
arr = []
hold = []

N = int(input("Enter the number of men: "))

for x in range(2 * N):
    arr.clear()
    for y in range(N):
        t = int(input("Enter: "))
        arr.append(t)
    hold = arr
    prefer.append(hold)

def wPrefersm1overm(prefer, w, m, m1):
    for i in range(N):
        if prefer[w][i] == m1:
            return True
        if prefer[w][i] == m:
            return False

wPartner = []
mFree = []

for x in range(N):
    wPartner.append(-1)

x = 0
for x in range(N):
    mFree.append(False)

free = N
while free > 0:#while there are free men
    m = 0
    for m in range(N):#choosing the first free man
        if mFree[m] == False:
            break
    for i in range(N):
        w = prefer[m][i]
        if wPartner[w - N] == -1:#if the woman is free
            wPartner[w - N] = m
            mFree[m] = True
            free = free - 1
        else:#if the woman is not free. Check whether that woman may change her mind for a better match.
            m1 = wPartner[w - N]#current partner of w
            if wPrefersm1overm(prefer, w, m, m1) == False:
                wPartner[w - N] = m
                mFree[m] = True#m has a partner now
                mFree[m1] = False#m1 do not have partner now


print("Woman    Man")
for j in range(N):
    print(i + N, "      ", wPartner[j])
```

Illustrating the example from Problem 1:

Step 1:

Each man proposes to the woman he most prefers:

-m1 proposes to w1

-m2 proposed to w1

-m3 proposes to w2

	w1	w2	w3
m1	X		
m2	X		
m3		X	

w1 receives proposals from m1 and m2. She chooses the proposal from m1 since she prefers m1 to m2.

	w1	w2	w3
m1	x		
m2			
m3		x	

Step 2:

Since m2 has been rejected by w1, he proposes his second choice w3.

	w1	w2	w3
m1	x		
m2			x
m3		x	

All women and men are matched.
Program terminates.

Result:
m1 - w1
m2 - w3
m3 - w2